

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of preventing buffer overrun security vulnerabilities comprising:
placing a return address on a stack;
~~executing a modified call routine for placing~~ adding a random amount plurality of empty space-spaces to a known place onto a ~~on the stack;~~
executing a called function; and
~~executing a modified return routine for removing said random amount one or~~
more of the plurality of empty space-spaces from the stack to find the
return address; and
setting an end of stack pointer to an end of stack frame.
2. (Currently Amended) The method of claim 1, wherein ~~said modified call routine comprises further comprising:~~
~~placing a return address for the called function on the stack;~~
~~calculating a random number;~~
~~saving said~~ the ~~random number in a secure location;~~
~~placing a plurality of blank bytes equal to the random number onto to the stack;~~
~~building a stack frame by placing values from the called function onto to the~~
~~stack; and~~
setting an end of stack pointer to an end of the stack frame.
3. (Currently Amended) The method of claim 2, wherein ~~said~~ the ~~location is~~
comprises a processor register that is not generally accessible.

4. (Currently Amended) The method of claim 1, ~~wherein said modified return routine comprises further compsing:~~
~~recalling a~~the random number saved during an execution of said modified call routine;
removing a number of bytes equal to said the random number from the stack;
retrieving ~~a~~the return address for the called function from the stack; and
setting an end of stack pointer to an end of a previous stack frame.
5. (Currently Amended) The method of claim 1, ~~wherein said modified call routine comprises~~further comprising:
placing a return address for the called function on the stack;
calculating a hash value of stack invariants;
saving said the hash value in a secure location; and
building a stack frame by placing values from the called function onto the stack.

Claims 6-11 (Cancelled)

12. (Currently Amended) An apparatus, comprising:
a storage device having stored therein one or more routines for preventing buffer overrun security vulnerabilities; and
a processor coupled to the storage device for executing the one or more routines that, when executing the routines, prevents buffer overrun errors by:

placing a return address on a stack;
~~executing a modified call routine for placing adding a random amount~~
plurality of empty space spaces to a known place onto a~~on the~~
stack;
executing a called function; and
~~executing a modified return routine for removing said random amount one~~
or more of the plurality of empty space spaces from the stack to
find the return address; and
setting an end of stack pointer to an end of stack frame.

13. (Currently Amended) The apparatus of claim 12, ~~wherein said modified call routine comprises further comprising:~~
~~placing a return address for the called function on the stack;~~
calculating a random number;
saving ~~said~~ the random number in a secure location;
placing a plurality of blank bytes equal to the random number ~~onto~~ to the stack;
building a stack frame by placing values from the called function onto the stack;
and
setting an end of stack pointer to an end of the stack frame.
14. (Currently Amended) The apparatus of claim 13, wherein ~~said~~-location is
comprises a processor register that is not generally accessible.

Claims 15-22 (Cancelled)

23. (Currently Amended) A machine-readable medium having stored thereon data representing sequences sets of instructions, ~~said sequences of instructions which, when executed by a processor~~ machine, cause said processor the machine to prevent buffer overrun errors by:
- place a return address on a stack;
- executing a modified call routine for placing adding a random amount plurality of empty space spaces to a known place onto on a stack;
- executing execute a called function; and
- executing a modified return routine for removing said random amount remove one or more of the plurality of empty space spaces from the stack to find the return address; and
- set an end of stack pointer to an end of stack frame.
24. (Currently Amended) The machine-readable medium of claim 23, wherein said modified call routine comprises: the sets of instructions which, when executed by the machine, further cause the machine to:
- placing a return address for the called function on the stack;
- calculating a random number;
- saving said the random number in a secure location;
- placing a plurality of blank bytes equal to the random number onto to the stack;
- building a stack frame by placing values from the called function onto the stack;
- and
- setting an end of stack pointer to an end of the stack frame.

25. (Currently Amended) The machine-readable medium of claim 24, wherein said
the location is comprises a processor register that is not generally accessible.

Claims 26-33 (Cancelled)

34. (New) A system, comprising:
- a storage medium; and
 - a processor coupled with the storage medium, the processor to
 - placing a return address on a stack,
 - adding a plurality of empty spaces to a known place on the stack,
 - executing a called function,
 - removing one or more of the plurality of empty spaces from the stack to
 - find the return address, and
 - setting an end of stack pointer to an end of stack frame.
35. (New) The system of claim 34, further comprising:
- calculating a random number;
 - saving the random number in a secure location;
 - placing a plurality of blank bytes equal to the random number to the stack;
 - building a stack frame by placing values from the called function onto the stack;
 - and
 - setting an end of stack pointer to an end of the stack frame.

36. (New) The system of claim 35, wherein location comprises a processor register that is not generally accessible.